ORC 67-21
June 1967

CYCLES WITH MINIMUM AVERAGE LENGTH

by

Alain Fillières

OPERATIONS RESEARCH CENTER

COLLEGE OF ENGINEERING

UNIVERSITY OF CALIFORNIA - BERKELEY

CYCLES WITH MINIMUM AVERAGE LENGTH

by

Alain Fillières
Attaché de recherche au CNRS (France)
and the Operations Research Center
University of California, Berkeley

June 1967 ORC 67-21

This research was supported by the U.S. Army Research Office-Durham under Contract DA-31-124-ARO-D-331 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

The idea developed in this paper is contained in the author's thesis [1]. My thanks to K. G. Murty for his valuable help.

ABSTRACT

Given a directed network whose arcs have lengths unrestricted in sign and which contains at least one cycle, an algorithm to find the minimum average length cycle (length divided by its number of arcs) is described. A direct application of this algorithm solves the problem of finding whether a directed graph contains a cycle with negative length.

CYCLES WITH MINIMUM AVERAGE LENGTH

by

Alain Fillières

1. Notations and Definitions

We denote a directed *graph* (N, Γ) by:

A finite set N of nodes i, $i \in \{1, 2, \ldots, n\}$. A mapping Γ : $N \to 2^N$ where 2^N is the set of all subsets of N. $j \in \Gamma(i)$ means that there exists an arc (i,j) going from node i to node j. We will denote by $\gamma(i)$ an element of $\Gamma(i)$; it will be useful to write
$\Gamma^{k+1}(i) = \Gamma(\Gamma^k(i))$, $k = 0$, 1, 2, ...

Let $A = \{(i,j)/i\epsilon N \text{ , } j\epsilon\Gamma(i)\}$ be the set of arcs. A mapping ℓ : $A \to \mathbb{R}$ is given, $\ell(i,j)$ is called the *length of arc* (i,j) . We define

$$\ell(K) = \sum_{a \epsilon K} \ell(a)$$

where K is any subset of A .

Note that lengths are unrestricted in sign.

We will assume in the following that:

(N, Γ) contains at least one cycle.

(N, Γ) is so that $\Gamma(i) \neq \phi$ $\forall i \epsilon N$. This is no real restriction since one can always add to the set A an arc (i,i) with a length $\ell(i,i) = \infty$ for every i which has $\Gamma(i) = \phi$, in the original graph.

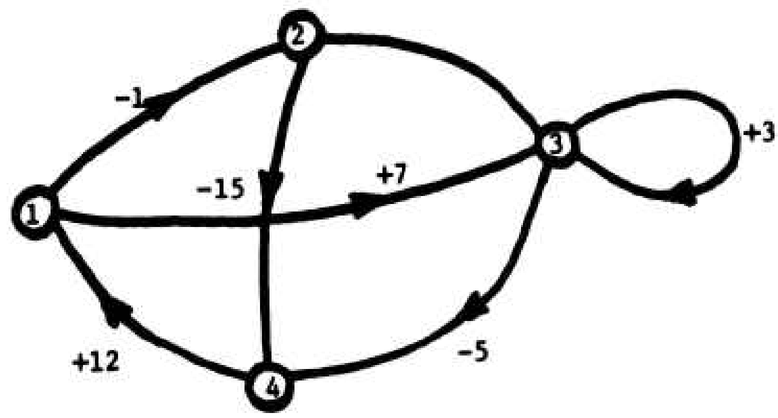
A *p-arcs cycle* is a set:

$C = \{i, \gamma(i), \gamma^2(i), \ldots, \gamma^p(i) = i\}$ where $\gamma^p(i)$ is an element of $\Gamma^p(i)$.

The *average length* of the p-arcs cycle C is defined by:

$$\bar{\ell}(C) = \frac{1}{p} \sum_{k=0}^{p-1} \ell(\gamma^k(i)\ ,\ \gamma^{k+1}(i))$$

Example:



Let $C = (1,3,4,1)$

$$\bar{\ell}(C) = \frac{7-5+12}{3} = \frac{14}{3}\ .$$

This paper describes an algorithm which yields *the minimum* average length cycle* in the graph (N, Γ) defined above. Thus this algorithm can also be used to find out whether the graph contains a negative length cycle. The method is an application to a deterministic case of a policy-iteration method for multiple Markov chain processes with rewards [2], [3] .

2. Policy γ

A *policy* γ *is a mapping* $\gamma : N \rightarrow N$ *that associates to every* $i \in N$ *an element* $\gamma(i) \in \Gamma(i)$.

* To get the maximum average length cycle, we replace the original arc lengths ℓ by $-\ell$ and apply the same algorithm.

Let (N,γ) be the subgraph of (N,γ) representing a policy γ. The set of arcs of (N,γ) is:

$$A_\gamma = \{(i,j)/i\epsilon N,\ j = \gamma(i)\}\ .$$

There are $\prod_{i\epsilon N} |\Gamma(i)|$ different policies with an equal number of associated graphs (N,γ), where $|\Gamma(i)|$ is the cardinality of the set $\Gamma(i)$.

3. Properties of Graphs (N,γ)

P.1 (N,γ) breaks down into connected components (N_h,γ_h), $h\epsilon\{1, 2, \ldots, k\}$ such that:

$$N_i \cap N_j = \phi \qquad i,j\epsilon\{1,2, \ldots, k\}$$

$$\bigcup_{h=1}^{k} N_h = N$$

P.2 (N_h, γ_h), $h\epsilon\{1, \ldots, k\}$ contains one and only one cycle.

P.3 Every $i\epsilon N_h$, $h\epsilon\{1, \ldots, k\}$ is either a node of the unique cycle of (N_h, γ_h), or a node of a unique chain leading to this cycle.

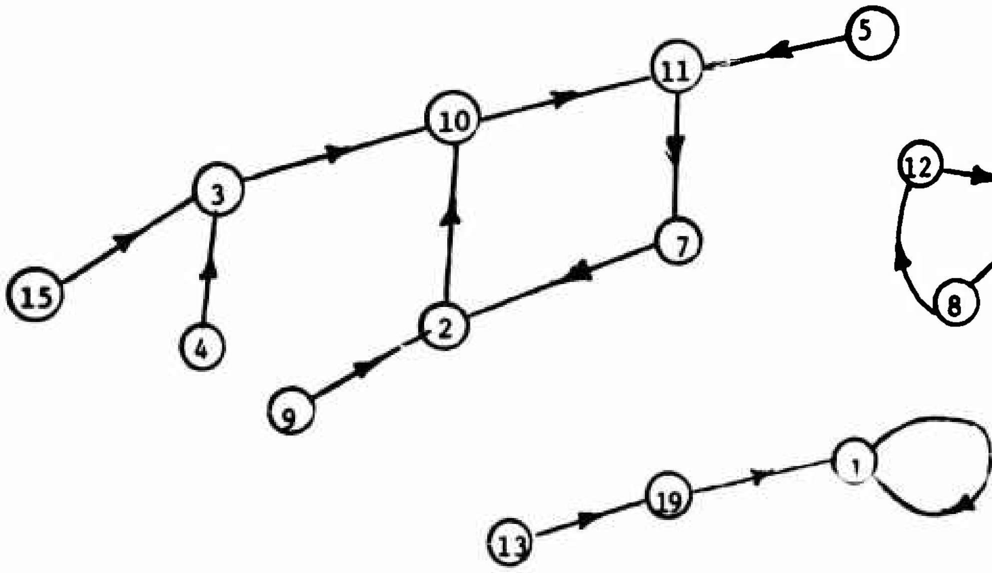
These properties are an obvious consequence of the definition of γ and the fact that N is a finite set.

k is the number of connected components of (N,γ).

There is a one-to-one correspondence between the set of cycles defined by γ, and the connected components of (N,γ).

Let us call C_i^γ the cycle in the connected component containing node i.

<u>**Example:**</u>



$N = \{1, 2, \ldots, 15\}$ $\gamma(1) = 1$

$\gamma(2) = 10$

$\gamma(3) = 10$

etc., ...

$$C_i^{\gamma} = \begin{cases} \{10,11,7,2,10\} & \text{for } i \in \{2,3,4,7,9,10,11,15\} \\ \{12,5,8,12\} & \text{for } i \in \{6,8,12\} \\ \{1,1\} & \text{for } i \in \{1,13,14\} \end{cases}$$

4. Policy Cost and Optimality

The cost vector of a policy γ is the n-vector g :

$$g = [g_1, g_2, \ldots, g_n]$$

when g_i is the average length of the cycle C_i^{γ} .

γ *is better than* γ' , $\gamma \prec \gamma'$, if $g \leq g'$.

$\hat{\gamma}$ *is optimal* if $\hat{\gamma} \prec \gamma$ for all γ .

γ *and* γ' *are equivalent* if and only if $\gamma(i) = \gamma'(i)$, $\forall i \in N$.

In other words, $C_i^{\hat{\gamma}}$ is the minimum average length cycle which can be reached from i in the initial graph (N,Γ) .* Therefore $(N,\hat{\gamma})$ contains the minimum average length cycle of (N,Γ) .

The principle of the policy - iteration procedure is to start with an arbitrary policy and to improve it step by step (according to the criterion $\prec$ described above) until optimality is reached.

5. Functional Relation

A policy γ being given, we have a first relation

$$(1) \qquad g_i = g_{\gamma(i)} \qquad \forall i \in N$$

where $\gamma(i)$ is associated with i by the policy γ .

Let $\lambda_i(t)$ be the length of a t-arcs chain starting from i in (N,γ) ; $\lambda_i(t)$ satisfies:

$$(2) \qquad \lambda_i(t+1) = \ell(i,\gamma(i))+\lambda_{\gamma(i)}(t) \qquad \forall i \in N .$$

---

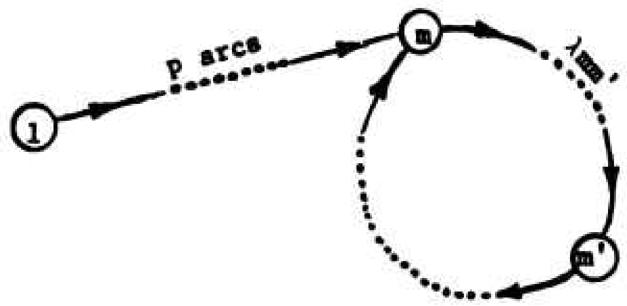
* It is important to note that we are not dealing with the problem of finding the minimum average length cycle which goes through a given node. An algorithm for this case would solve the Traveling Salesman Problem by adding an arbitrary large number to the length of every arc leaving this node.

If t is sufficiently large, from properties P_1, P_2, P_3 one can see that starting from any node i and coming along a large number of arcs one goes in general:

a) along a unique chain up to a node, say m ,

b) a large number of times around the cycle starting from m ,

c) stop at a node of the cycle, say m' .



So $\lambda_i(t)$ can be expressed in the following form:

$$\lambda_i(t) = \lambda_i(p) + \left[\frac{t-p}{r}\right]\ell(C)+\lambda_{mm'}$$

when

$p \geq 0$ is the number of arcs between i and m , which corresponds to (a)

$r =$ the number of arcs of the cycle C_i^γ

$\lambda_{mm'}$ the distance between m and m' along C_i^γ .

For a large value T of t

$$\left[\frac{T-p}{r}\right]\ell(C) \sim \frac{T}{r}\,\ell(C)$$

or T times the average length of C_i^γ . *Besides the quantity:* $\lambda_i(p) + \lambda_{mm}$, *that we will call* w_i *depend only on* i for a fixed T .

Hence, we get the relation:

$$\lambda_i(T) = Tg_i + w_i \quad \text{for} \quad T \text{ large .} \tag{3}$$

By using (3) in (2), we get:

$$(T+1)g_i + w_i = \ell(i,\gamma(i)) + Tg_{\gamma(i)} + w_{\gamma(i)}$$

or using (1)

$$g_i + w_i = \ell(i,\gamma(i)) + w_{\gamma(i)} \;.$$

Then, we get the following relations satisfied by any policy γ :

$$g_i = g_{\gamma(i)} \tag{4}$$

$$g_i + w_i = \ell(i,\gamma(i)) + w_{\gamma(i)} \tag{5}$$

for all $i \in N$.

(4) and (5) give n equations in the $n+k$ unknown variables $g_1, \ldots, g_n$ and $w_1, \ldots, w_k$. However, we only need to know the value of the differences $w_i - w_j$ for i, j in the same connected component. So we set $w_{i_h} = 0$ for an arbitrary i_h in each connected component, $h = 1,2, \ldots, k$, and we call v_i *the relative value of* w_i obtained by this way.

The value g_i and v_i associated with every node $i \epsilon N$ for a given policy, are used to determine a better policy than γ.

6. Algorithm

Initial policy γ_o

γ_o can be arbitrarily choosen. A good starting policy is γ_o satisfying:

$$\ell(i,\gamma_o(i)) = \underset{h\epsilon\Gamma(i)}{\mathrm{Min}} [\ell(i,h)] \qquad \forall i\epsilon N$$

Let γ_k be the policy choosen at step $k = 0,1,2, \ldots$ and $g_k(i)$, $v_k(i)$, $\forall i\epsilon N$, the values defined in Section 5 corresponding to γ_k.

Iteration

The step $k+1$ which yields a better polciy γ_{k+1} proceeds in two phases.

(I) Solve the following system for $g_k(i)$ and $v_k(i)$:

$$\left\{\begin{array}{ll} g_k(i) = g_k(\gamma_k(i)) & (6) \\ g_k(i) + v_k(i) = \ell(i,\gamma_k(i)) + v_k(\gamma_k(i)) . & (7) \end{array}\right. \qquad i = 1, \ldots, n$$

Note that in practice the set of equations (6) is not necessary, a policy γ_k being given it is easy to compute directly the value $g_k(i)$ for $i = 1, \ldots, n$ and replace them in (7).

(II) γ_{k+1} is obtained by letting $\gamma_{k+1}(i)$, $i = 1, \ldots, n$, in the following way:

Let h be satisfying:

(8) $$g_k(h) = \min_{j\epsilon\Gamma(i)} (g_k(j)) .$$

<u>Case 1:</u>

If h is unique choose:

$$\gamma_{k+1}(i) = h .$$

<u>Case 2:</u>

If h is not unique, choose an arbitrary $\gamma_{k+1}(i)$ satisfying:

(9) $$\ell(i,\gamma_{k+1}(i)) + v_k(\gamma_{k+1}(i)) = \min_{j\epsilon\Gamma(i)} (\ell(i,j) + v_{k+1}(j)) .$$

In both Case 1 and Case 2, use the following rule (R):

(R) <u>If $\gamma_k(i)$ satisfies (8) and (9), set $\gamma_{k+1}(i) = \gamma_k(i)$</u> .

The rule (R) means that if the node associa ed to i in the k^{th} step satisfies (8) and (9) that will be the node associated to i in the $k+1^{th}$ step.

Note that if (N,γ_k) has only one connected component, test (8) can be skipped.

The tests (8) and (9) applied to every node i, $i = 1,2, \ldots, n$, give a new policy γ_{k+1} . At this stage, there are two possibilities:

(a) $\gamma_{k+1}(i) = \gamma_k(i)\ \forall i \in N$, then $\gamma_k = \hat{\gamma}$

(b) $\exists i$ *such that* $\gamma_{k+1}(i) \neq \gamma_k(i)$, then go back to (I) with $k = k + 1$.

Remarks

1) The test (1) implies that the comparisons of the quantities $\ell(i,j) + v_k(j)$ are made with j belonging to the same connected component, hence, the sense of the relative values: v_k of the w_k .

2) If (N,Γ) is strongly connected, $\hat{\gamma}$ is such that $(N,\hat{\gamma})$ contains only one connected component and hence has a unique minimum average length cycle.

3) At each step of the procedure, in particular at the last one, one knows the immediate descendent of each node, hence, it is easy to get the cycles.

7. Proof of the Algorithm

We need to prove the two following statements:

1) If for any i belonging to any cycle of (N,γ_{k+1}) $\gamma_k(i) \neq \gamma_{k+1}(i)$ then $g_{k+1}(i) < g_k(i)$. (From the properties of (N,λ_{k+1}) that implies $g_{k+1} \leq g_k(i)$ $\forall i \in N)$.

2) If $\gamma_k(i) = \gamma_{k+1}(i)$ $\forall i \in N$ then $\gamma_k = \hat{\gamma}$. That means that when the procedure stops, one cannot find a policy which leads to a better value of the minimum $g(i)$, $\forall i \in N$. (Rule (R) implies the procedure stops on a finite number of steps.)

Proof 1:

Let's assume $\gamma_k(i_o) \neq \gamma_{k+1}(i_o)$ for at least one i_o belonging to a cycle

of (N, γ_{k+1}) . The tests (1) and (2) of phase II implies:

(10) $$g_k(\gamma_{k+1}(i)) - g_k(\gamma_k(i)) = \psi_i \qquad \forall i \in N$$

$$\text{where} \quad \psi_i \leq 0$$

(11) $$\ell(i, \gamma_{k+1}(i)) + v_{k+1}(\gamma_{k+1}(i)) - \ell(i, \gamma_k(i)) - v_k(\gamma_k(i)) = \phi_i \qquad \forall i \in N$$

$$\text{where} \quad \phi_i \leq 0 .$$

Note that from the Rule (R):

$$\gamma_{k+1}(i) = \gamma_k(i) \Rightarrow \phi_i = 0$$

$$\gamma_{k+1}(i) \neq \gamma_k(i) \Rightarrow \phi_i < 0$$

Phase I leads to the systems:

$$\begin{cases} g_k(i) = g_k(\gamma_k(i)) & (12) \\ g_k(i) + v_k(i) = \ell(i, \gamma_k(i)) + v_k(\gamma_k(i)) & (13) \end{cases}$$

$$\begin{cases} g_{k+1}(i) = g_{k+1}(\gamma_{k+1}(i)) & (14) \\ g_{k+1}(i) + v_{k+1}(i) = \ell(i, \gamma_{k+1}(i)) + v_{k+1}(\gamma_{k+1}(i)) & (15) \end{cases}$$

By introducing: ψ_i , ϕ_i , $\Delta g_k(i) = g_{k+1}(i) - g_k(i)$, $\Delta v_k(i) = v_{k+1}(i) - v_k(i)$,

in the differences (14) - (12) and (15) - (14), we get:

(16) $$\Delta g_k(i) = \psi_i + \Delta g_k(\gamma_{k+1}(i))$$

(17) $$\Delta g_k(i) + \Delta v_k(i) = \phi_i + \Delta v_k(\gamma_{k+1}(i))$$

for all $i \epsilon N$; ψ_i , $\phi_i \leq 0$

Let C_h be the r-cycle of the h^{th} connected component of (N, γ_{k+1}) which contains i_o satisfying the hypothesis, i.e., $\phi_{i_o} < 0$.

By adding (16) for $i \epsilon C_h$:

$$\sum_{i \epsilon C_h} \Delta g_k(i) = \sum_{i \epsilon C_h} \psi_i + \sum_{i \epsilon C_h} \Delta g_k(\gamma_{k+1}(i))$$

Then, $\sum_{i \epsilon C_h} \psi_i = 0$ which implies $\psi_i = 0$, $\forall i \epsilon C_h$.

Hence:

(18) $$\Delta g_k(i) = \Delta g_k(\gamma_{k+1}(i)) \qquad \forall i \epsilon C_h .$$

By adding (17) for $i \epsilon C_h$ and using (18), we get

$$r \Delta g_k(i) = \sum_{i \epsilon C_h} \phi_i \qquad \phi_i \leq 0 .$$

From the hypothesis, there exists $i_o \varepsilon C_h$ such that $\phi_{i_o} < 0$ the R. H. S. of (10) is negative which implies:

$$\Delta g_k(i) < 0 \qquad \forall i \varepsilon C_h \ .$$

This proof applies for all h satisfying the hypothesis, Q. E. D .

Proof 2:

Let $\hat{\gamma} = \gamma_s$ be the policy obtained at the end of the procedure. Let us suppose there exists a policy γ_t such that:

$$g_t(i_o) < g_s(i_o) \quad \text{for at least one } i_o$$

since, γ_t did not come out from the procedure, we have:

$$\text{(10')} \qquad g_s(\gamma_t(i)) - g_s(\gamma_t(i) = \eta_i \qquad \eta_i \geq 0$$

$$\text{(11')} \qquad v_s(\gamma_t(i)) + \ell(i,\gamma_t(i)) - v_s \ (\gamma_s(i)) - \ell \ (i,\gamma_s(i)) = \mu_i \qquad \mu_i \geq 0$$

for all $i \varepsilon N$.

Besides, we get the equivalent relation as in Proof 1, with $k = s$, $k + 1 = t$. Let

$$\begin{cases} \Delta_{ts}g(i) = g_t(i) - g_s(i) \\ \Delta_{ts}v(i) = v_t(i) - v_s(i) \end{cases}$$

(16') $$\Delta_{ts}g(i) = \eta_i + \Delta_{ts}g(\gamma_t(i))$$

(17') $$\Delta_{ts}g(i) + \Delta_{ts}v(i) = \mu_i + \Delta_{ts}v(\gamma_t(i))$$

$$\eta_i \geq 0 \ , \ \mu_i \geq 0 \ .$$

Let C'_h be the p'- cycle of (N,γ_t) which contains i_o ; by adding (16') and (17') for $i \varepsilon C'_h$, we get as in Proof 1:

$$p'\Delta_{ts}g(i) = \sum_{i\varepsilon C'_h} \mu_i \qquad\qquad \mu_i \geq 0$$

then,

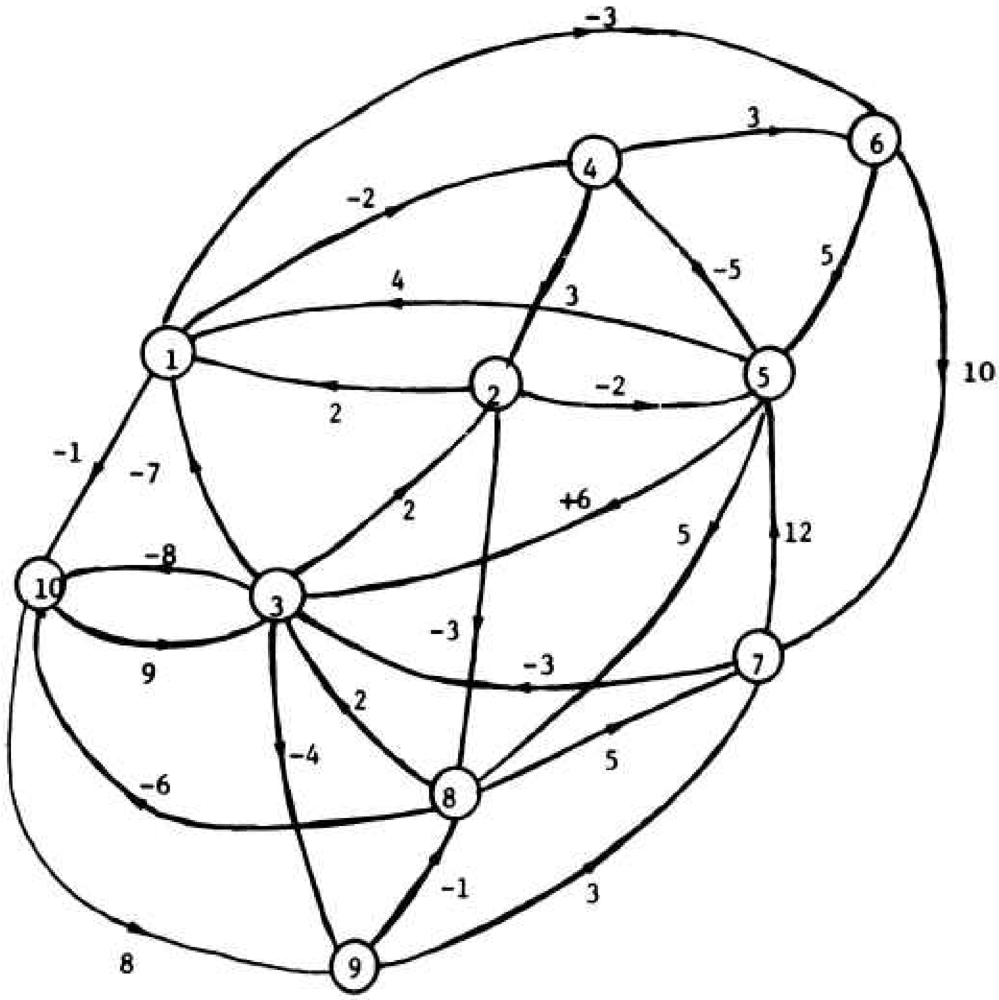
$$\Delta_{ts}g(i) > 0$$

since

$$\mu_{i_o} > 0 \ .$$

Hence, the contradiction.

8. Example:



The value of $\gamma_k(i)$, $g_k(i)$, $v_k(i)$, $k = 0, 1, 2, 3, 4$, $i = 1,2, \ldots, 10$, are given in the following tableau (computations have been made by hand).

	γ_0	g_0	v_0	γ_1	g_1	v_1	γ_2	g_2	v_2	γ_3	g_3	v_3	γ_4	g_4	v_4	γ_5
1	6	2	-5	10	$-\frac{4}{3}$	$\frac{1}{3}$	6	$-\frac{4}{3}$	$-\frac{7}{3}$	4	$-\frac{4}{3}$	$-\frac{22}{3}$	4	-2	-6	4
2	8	$\frac{1}{3}$	$-\frac{29}{3}$	8	$-\frac{4}{3}$	$-\frac{27}{3}$	8	$-\frac{4}{3}$	$-\frac{26}{3}$	8	$-\frac{4}{3}$	$-\frac{26}{3}$	8	-2	-8	8
3	10	$\frac{1}{3}$	$-\frac{25}{3}$	9	$-\frac{4}{3}$	$-\frac{36}{3}$	9	$-\frac{4}{3}$	$-\frac{31}{3}$	9	$-\frac{4}{3}$	$-\frac{31}{3}$	1	-2	-11	1
4	5	2	-10	2	$-\frac{4}{3}$	$-\frac{14}{3}$	5	$-\frac{4}{3}$	$-\frac{20}{3}$	5	$-\frac{4}{3}$	$-\frac{20}{3}$	5	-2	-6	5
5	1	2	-3	3	$-\frac{4}{3}$	$-\frac{14}{3}$	3	$-\frac{4}{3}$	$-\frac{9}{3}$	3	$-\frac{4}{3}$	$-\frac{9}{3}$	3	-2	-3	3
6	5	2	0	7	$-\frac{4}{3}$	$-\frac{36}{3}$	7	$-\frac{4}{3}$	$-\frac{2}{3}$	7	$-\frac{4}{3}$	$-\frac{2}{3}$	7	-2	0	7
7	3	$\frac{1}{3}$	$-\frac{35}{3}$	3	$-\frac{4}{3}$	$-\frac{41}{3}$	3	$-\frac{4}{3}$	$-\frac{21}{3}$	3	$-\frac{4}{3}$	$-\frac{21}{3}$	3	-2	12	3
8	10	$\frac{1}{3}$	$-\frac{19}{3}$	7	$-\frac{4}{3}$	$-\frac{22}{3}$	3	$-\frac{4}{3}$	$-\frac{21}{3}$	3	$-\frac{4}{3}$	$-\frac{21}{3}$	3	-2	-7	3
9	8	$\frac{1}{3}$	$-\frac{23}{3}$	7	$-\frac{4}{3}$	$-\frac{28}{3}$	7	$-\frac{4}{3}$	$-\frac{23}{3}$	7	$-\frac{4}{3}$	$-\frac{23}{3}$	7	-2	-7	7
10	9	$\frac{1}{3}$	0	9	$-\frac{4}{3}$	0	3	$-\frac{4}{3}$	0	3	$-\frac{4}{3}$	0	3	-2	0	3

$\gamma_4 = \gamma_5 = \hat{\gamma}$

The minimum average length cycle is $(1,4,5,3,1)$; its average length is -2 . This cycle is unique, $\gamma_4 = \gamma_5$ has only one connected component.

Remarks:

1) γ_1 shows that the original graph has at least a negative length cycle.

2) We set arbitrarily $v_o(6) = v_o(10) = 0$ components of γ_o , and $v_k(10) = 0$, $k = 1,2,3,4$.

3) γ_2 , γ_3 did not improve the value of g because the nodes of the cycle $(3,9,7,3)$ kept the same immediate descendents during Steps 2 and 3 .

REFERENCES

[1] Fillières, A., "Les Chemins d'un graphes," These de 3^{o} cycle. Faculte des Sciences de l'Universite de Paris, (1966).

[2] Howard, R. A., DYNAMIC PROGRAMMING AND MARKOV PROCESSES, Ed. Technology Press and Wiley, New York, (1960).

[3] Rosenstiehl, P. et Ghouila-Houri A., LES CHOIX ÉCONOMIQUES - DÉCISIONS SÉQUENTIELLES ET SIMULATION. Dunod, Paris, (1960).

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) University of California, Berkeley	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP NA
3. REPORT TITLE CYCLES WITH MINIMUM AVERAGE LENGTH	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Technical Report	
5. AUTHOR(S) (Last name, first name, initial) Fillieres, Alain	

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1967	18	3

8a. CONTRACT OR GRANT NO. DA-31-124-ARO-D-331 b. PROJECT NO. 20014501B14C c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) ORC 67-21 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) 5307.24-M
10. AVAILABILITY/LIMITATION NOTICES Distribution of this report is unlimited.	
11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY U.S. Army Research Office-Durham Box CM, Duke Station Durham, North Carolina 27706

13. ABSTRACT

Given a directed network whose arcs have lengths unrestricted in sign and which contains at least one cycle, an algorithm to find the minimum average length cycle (length divided by its number of arcs) is described. A direct application of this algorithm solves the problem of finding whether a directed graph contains a cycle with negative length.

14. KEY WORDS

arcs
algorithm
cycles

DD FORM 1 JAN 64 1473

Unclassified
Security Classification